Министерство образования и науки Российской Федерации

МГТУ им .Н.Э Баумана

Факультет «Робототехника и комплексная автоматизация»

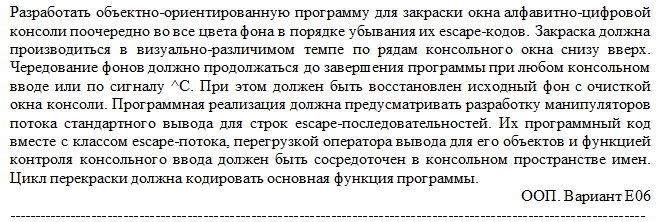
Кафедра «РК6»

**Отчет по лабораторной работе**

По курсу «Объектно-ориентированное программирование»

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  |  |  |  |  |
| Выполнил: |  |  | Студент | Журавлев Н.В. |
|  |  |  | Группа | РК6-22Б |
|  |  |  |  |  |
| Проверил: |  |  |  | Кузьмина И.А. |
|  |  |  | Дата | 28.05.2020 |
|  |  |  | Подпись | \_\_\_\_\_\_\_\_\_\_\_ |

**Домашнее задание №2**



**Описание алгоритма**

1. Делаем цвет курсива синим
2. Делаем фон бесцветным
3. Ширину курсора устанавливаем 1
4. Перемешаем курсор на последнюю строку и очищаем её
5. Включаем мигание и на последнюю строку пишем надпись ^C or Enter to exit и выключаем мигание
6. Выводим на предпоследнюю строчку размеры терминала
7. Пока не будет вводу, для каждой строки добавляется отдельный новый в порядке убывания escape-кодов цвет, заполняя одну строчку.
8. Очистка терминала и возрат к исходному состоянию

**Текст программы**

#include <iostream>

#include <sstream>

#include <iomanip>

#include <sys/ioctl.h>

#include <termios.h>

#include <fcntl.h>

#include <stdio.h>

#include <stdlib.h>

#include <signal.h>

#include <unistd.h>

static int done = 0;

using namespace std;

namespace con {

int comax() {

struct winsize w;

ioctl(0, TIOCGWINSZ, &w);

return(w.ws\_col);

}

int romax() {

struct winsize w;

ioctl(0, TIOCGWINSZ, &w);

return(w.ws\_row);

}

ostream& ED(ostream& s) {

return s << string("\033[2J");

}

ostream& EL(ostream& s) {

return s << string("\033[K");

}

class estream {

private:

string escape;

public:

estream(string e) : escape(e) {};

friend ostream& operator<<(ostream&, estream);

};

ostream& operator<<(ostream& s, estream e) {

s << e.escape << flush;

return s;

}

estream CUP(int y, int x) {

ostringstream sout;

sout << "\033[" << y << ";" << x << "H";

return estream(sout.str());

}

estream SGR(int r) {

ostringstream sout;

sout << "\033[" << r << "m";

return estream(sout.str());

}

}

using con::SGR;

using con::CUP;

void interruptor(int signo) {

done = signo;

return;

}

int kbin() {

char buf[512];

int n=0;

int flags = fcntl(0, F\_GETFL);

usleep(1);

fcntl(0, F\_SETFL, flags | O\_NONBLOCK);

n = read(0, buf, 512);

fcntl(0, F\_SETFL, flags );

return(n);

}

int kbhit() {

int n = 0;

char buf[512];

struct termios t[2];

tcgetattr(0, &t[0]);

tcgetattr(0, &t[1]);

t[0].c\_lflag &= ~(ICANON | ECHO);

t[0].c\_cc[VMIN] = 0;

t[0].c\_cc[VTIME] = 0;

tcsetattr(0, TCSANOW, &t[0]);

n = read(0, buf, 512);

tcsetattr(0, TCSAFLUSH, &t[1]);

return(n);

}

int main() {

int m;

cout << SGR(30+9);

cout << SGR(40+9);

cout << SGR(1);

cout << con::CUP(con::romax(), 1);

cout << con::ED;

cout << con::SGR(5);

cout << "^C or Enter to exit" << flush;

cout << con::SGR(20 + 4);

cout << con::CUP(con::romax() - 1, 1);

cout << con::romax() << "x" << con::comax() << flush;

int x, y,flag = 0;

char a;

int f, b;

srand(getpid());

signal(SIGINT, interruptor);

a = 32; b = 48; f = b-10;

while(done < 1) { // while no ^C

for ( y = con::romax() - 2; y > 0; y--){

b--;

f=b-10;

if(b < 40){

b = 47;

f = b - 10;

}

for (x = con::comax(); x > 0; x--){

cout << SGR(b) << CUP(y, x);

cout << SGR(f) << a << flush;

}

usleep(250000);

if(kbhit() > 0 || done > 1){

flag = 1;

break;

}

}

b = 48;

if(flag)

break;

}

cout << CUP(38, 142) << SGR(0) << con::EL;

{

int x, y;

char a;

struct termios t[2];

tcgetattr(0, &t[0]);

tcgetattr(0, &t[1]);

t[0].c\_lflag &= ~(ICANON | ECHO);

tcsetattr(0, TCSAFLUSH, &t[0]);

cout << "\033[512;512H" << flush;

cout << "\033[6n" << flush;

cin >> a >> a >> y >> a >> x >> a;

cout << " info:" << y << ';' << x << endl;

tcsetattr(0, TCSAFLUSH, &t[1]);

}

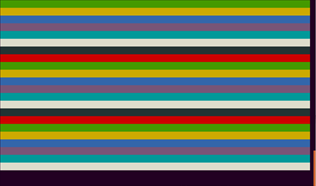
cout << "\033[2J\033[1;1H" << flush;

return(m);

}

**Тесты**

**Вывод**



**Результат**

Перекрашенный терминал

**Списки литературы**

1)Волосатова Т. М., Родионов С. В. «Объектно-ориентированное программирование на С++» 2)http://bigor.bmstu.ru/?cnt/?doc=VU/base.cou